



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 0 952 519 A1**

(12) **EUROPÄISCHE PATENTANMELDUNG**

(43) Veröffentlichungstag:
27.10.1999 Patentblatt 1999/43

(51) Int. Cl.⁶: **G06F 9/46**

(21) Anmeldenummer: **98103827.6**

(22) Anmeldetag: **04.03.1998**

(84) Benannte Vertragsstaaten:
**AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SE**
Benannte Erstreckungsstaaten:
AL LT LV MK RO SI

(71) Anmelder:
**SIEMENS AKTIENGESELLSCHAFT
80333 München (DE)**

(72) Erfinder:
Die Erfindernennung liegt noch nicht vor

(54) **Plattform eines Rechnersystems**

(57) Aufgabe der Erfindung ist es, eine Plattform eines Rechnersystems anzugeben, die eine situationsgerechte Anpassung von Parameterwerten für die Lastverteilung und Laststeuerung durchführen kann. Diese Aufgabe wird durch eine plattformeinheitliche

Manager-Komponente gelöst, die alle Parameter, die die Lastverteilung und Laststeuerung und somit die Performance des Systems beeinflussen, verwaltet.

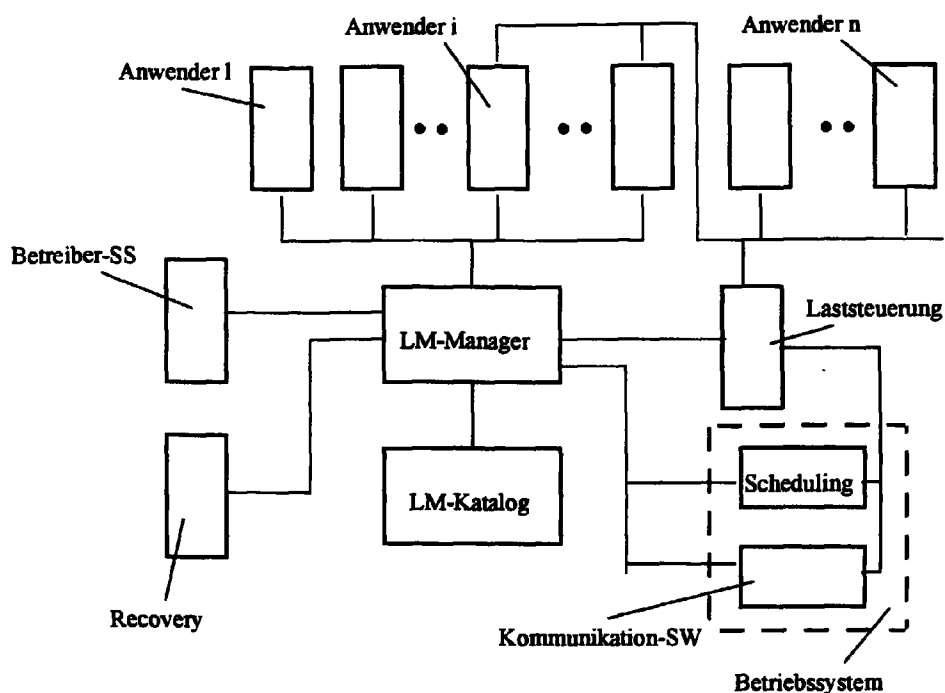


FIG 3

EP 0 952 519 A1

Beschreibung

[0001] Ein Vermittlungsknoten ist ein Rechner, von dem erwartet wird, daß er im Rahmen von Delay-Zeiten, die von den Kunden und den internationalen Standards gefordert werden, Aufträge bearbeiten kann. Beim Breitband-Vermittlungsknoten liegen diese Aufträge insbesondere auf sehr verschiedenen Gebieten, mit verschiedenen Delayzeit-Anforderungen, zeitlich schwankendem und ganz verschiedenartigem Lastaufkommen. Die Aufträge kommen außer aus der Vermittlungstechnik für Daten- und Sprachverkehr aus Aufgabenbereichen wie den Protokoll-Bearbeitungen (z.B. CCS7-Signalisierung, Breitband UNI-Interface, Breitband NNI-Signalisierung, Global Title Translation, PNNI-Protokoll, Public Access Interface, Private Branch Exchange Access, ...), dem Management der Signallisierungsnetze (insbesondere bei Netzausfällen), der Bedienung der Betreiberschnittstelle, dem Billing, der Verkehrsstatistik, der Verwaltung der Teilnehmeranschlüsse, der Leitungsanschlüsse, der Wegedaten und Tarifzonen, weiterhin der Fehlerbehandlungen, die dafür sorgen, daß die „downtime“ (Zeit, in der der gesamte Vermittlungsknoten, oder Teilkomponenten, nicht verfügbar sind) minimiert wird.

[0002] Da dieser Rechner im Netzverbund arbeitet, müssen nicht nur der Normalbetrieb, sondern auch andere Betriebszustände, z.B. die Inbetriebnahme, bei der eine netzweite Kommunikation angestoßen wird, zeitlich optimal verlaufen. Abhängig von den Betriebszuständen liegen also die Delayzeit-Anforderungen verschieden, und es muß darauf adäquat reagiert werden.

[0003] Das von Vermittlungsknoten erwartete Realzeitverhalten stellt also spezielle Anforderungen an die Knotenarchitektur und das Betriebssystem. Diese bestehen darin, daß die Prozessorlaufzeitkapazitäten zu jeder Zeit bestmöglich auf die verschiedenen Anwendungsprogramme verteilt sein müssen.

[0004] In dem neuen ATM-Vermittlungsknoten werden, um das zu gewährleisten, dem Betriebssystem Zeitbudgetwerte zur Verfügung gestellt, so daß es den verschiedenen Applikationen eine Mindestzugriffszeit zum Prozessor garantieren kann.

[0005] Da Telekommunikationsnetze einem ständigem Wandel bezüglich Leistungsmerkmalen, Architektur und Größe unterworfen sind, müssen die Vermittlungsknoten in Telekommunikationsnetzen in den Funktionalitäten, die sie anbieten, ausgesprochen variabel und in ihrer Größe skalierbar sein.

[0006] Ein Vermittlungsrechner muß also insbesondere wachsen können, und zwar mit minimalen, also möglichst keinen Betriebsunterbrechungen: Dazu gehört der HW-Upgrade zur Verstärkung der bereits existierenden Prozessorleistung, der SW-Upgrade zum Hochrüsten der Leistungsmerkmale, und (unter Umständen im Zusammenhang damit) eine Neuverteilung der Zeitbudgets am Prozessor inkl. anderer Parameter, z.B. Lastschwellen für die Lastkontrolle,

Scanintervalle, Protokoll-Budgets. Insbesondere die genannten Parameter sollten unterbrechungsfrei änderbar sein.

[0007] Bisher erfolgte eine verteilte Ablage der genannten Parameter bei den verschiedenen Anwendern, d.h. die Parameter wurden in den Anwendungsprogrammen deklariert. Die Schnittstellenverwaltung bezüglich der Parameter erfolgte ebenfalls durch die Anwender.

[0008] Da die Gesamtheit der Laststeuerungsparameter und Lastverteilungsparameter (im folgenden kurz als Laststeuerungsparameter bezeichnet) in vierselliger Größenordnung liegen und sie für Lastverteilung (Zeitaufteilung auf die verschiedenen Applikationen entsprechend Durchsatz und Delayzeitanforderungen, von denen viele, aber nicht alle für das Scheduling gebraucht werden.) und Laststeuerung elementar wichtig sind, würde ihre (bisherige) verteilte Ablage bei den verschiedenen Anwendern und ihre Verwaltung durch sie (über entsprechend viele Anwender-Schnittstellen) zu einem erheblichen, wenn nicht unlösbaren Engineering-Problem führen, wobei das Problem darin bestehen würde, eine situationsgerechte Anpassung (bei der Konfiguration und zur Laufzeit) der aktuell gültigen Parameterwerte zum Zwecke der optimalen Lastverteilung und Laststeuerung durchzuführen.

[0009] Der Erfindung liegt die Aufgabe zugrunde, das genannte Problem zu lösen.

[0010] Die Lösung wird durch einen Last-Modell-Manager erreicht, der die Parameter, die die Laststeuerung und die Lastverteilung des Systems und somit die Performance des Systems beeinflussen, einheitlich in einem sogenannten Last-Modell-Katalog verwaltet.

[0011] Im folgenden wird ein Ausführungsbeispiel der Erfindung anhand der Zeichnung näher beschrieben, wobei die Zeichnung vier Figuren umfaßt.

[0012] Zunächst wird anhand von FIG 1 die Architektur eines bsph. ATM-Knotens dargelegt.

[0013] Die wichtigste Steuerung des Breitbandknotens wird in einem skalierbaren Cluster von Hauptprozessoren durchgeführt. Alle Prozessoren kommunizieren über ein ATM-Koppelnetz (ATM-Fabric) mittels eines gemeinsamen internen Protokolls. Ein einzelner Hauptprozessor mit seinem Anteil an der Gesamtsoftware des Systems wird im folgenden als Plattform bezeichnet.

[0014] Das hier betrachtete Cluster kann in seiner Maximalkonfiguration aus 128 einzelnen Plattformen bestehen. Hardware und Basis-Software (z.B. Betriebssystem und Laststeuerungssoftware) sind für alle gleich. Das heißt, diese Basis muß den Ansprüchen aller bekannter Nutzungsarten/Applikationen entsprechen. Zugleich muß sie auch so umfassend sein, daß jederzeit einer Forderung nach neuen (u.U. heute noch unbekannten) Nutzungen effizient nachgekommen werden kann. Diese Nutzungsarten sind durch verschiedenen Applikationssoftwarepaketen auf den Plattformen realisierbar. Es gibt Plattformen für:

- Vermittlungstechnik (Call Processing) (können mehrfach im Knoten vorhanden sein)
- Protokollumsetzungen (können mehrfach im Knoten vorhanden sein)
- Operation, Administration und Maintenance (können mehrfach im Knoten vorhanden sein)
- Management der Netzsignalisierung (ist genau einmal im Knoten vorhanden)
- Betreiberschnittstelle (ist genau einmal im Knoten vorhanden)

[0015] Die Gesamtheit der auf einer Plattform vorhandenen Software für o.g. Aufgaben einschließlich Basissoftware wird auch als Ladetyp bezeichnet. Über diese "reinen" Ladetypen hinaus müssen auch Plattformen mit sog. kombinierten Ladetypen bereitgestellt werden, damit Minimalkonfigurationen kundengerecht und kostengünstig bereitgestellt werden können. Gerade diese Plattformen sind eine Herausforderung für die sorgfältige Abstimmung der garantierten Minimallaufzeitbudgets der Applikationen, denn hier ist die Konkurrenz besonders verschiedenartiger Anwenderprogramme um die Zuteilung der Prozessorlaufzeit am größten.

[0016] Im folgenden wird das Betriebssystem und die Lastkontrolle näher erläutert.

Prozeßbegriff und CHILL

[0017] Die oben erwähnten Applikationen sind als Prozesse realisiert und werden im folgenden als solche referenziert. Als Prozeß wird hier eine laufzeitbezogene Einheit verstanden. Der Prozeß als ablauffähige Softwareeinheit hat Attribute, von denen hier diejenigen, die sein Zugangsrecht zum Prozessor beeinflussen, interessant sind. Die Attribute sind statisch implementiert. Der Prozeß mit seinen Attributen ist sprachlich (CHILL) erklärt und compilerunterstützt. Beim Systemhochlauf werden die Prozesse in einer betriebssystemgerechten Form, unter Berücksichtigung ihrer Attribute, bereitgestellt.

[0018] Die zwei für das Realzeitverhalten wichtigen Attribute der Prozesse bestimmen

- ☒ die Zugehörigkeit zu einer Aufgabengruppe und
- ☒ die Zugehörigkeit zu einer Kategorie von Delayzeit-Anforderungen innerhalb dieser Aufgabengruppe.

[0019] Die Aufgabengruppen sind diejenigen Einheiten, denen eine Mindestzugriffszeit auf den Prozessor der Plattform, auf der sie residieren, garantiert wird.

[0020] Die Kategorie der Delayzeitanforderungen eines Prozesses innerhalb dieser Aufgabengruppe bestimmt die Art, wie Prozesse, die auf der gleichen Plattform zur gleichen Aufgabengruppe gehören, um Rechenzeit konkurrieren, wenn sie gleichzeitig ablauf-

bereit sind. Dieses Attribut, das die Prozesse innerhalb einer Aufgabengruppe unterscheidet, ist die sog. Prozeßpriorität. (Vgl. unten den Abschnitt "Scheduling")

Betriebssystemanteil: Service Addressing

[0021] Besonders bei einem großen Ausbau des Knotens will man die Anforderungen so auf das Cluster verteilen, daß möglichst keine einzelne Plattform überfordert wird. Das übernimmt der Betriebssystemanteil Service Addressing.

[0022] Das Betriebssystem einer jeden Plattform kennt alle im gesamten Cluster angebotenen und ansprechbaren Funktionen als "Services". Ein Service kann auf der Plattform, auf der er verlangt wird, angeboten sein und/oder auf einer oder mehreren anderen Plattformen des Clusters. Um den Kommunikations-Overhead zwischen den Plattformen zu vermeiden, wird dem nachfragenden User die Ansprechadresse des Services der eigenen Plattform - falls dort vorhanden - mitgeteilt. Befindet sich diese Plattform jedoch in einem Überlastzustand (siehe auch unten den Abschnitt "Lastkontrolle"), so wird eine andere Plattform im Cluster ausgewählt, die diesen Service auch anbietet, sofern sich eine solche in einem niedrigeren Lastzustand befindet. Das ist möglich, weil sich auf jeder Plattform ein aktuelles Lastabbild des gesamten Clusters befindet.

Betriebssystemanteil: Scheduling

[0023] Besonders bei einer minimalen Konfiguration des Clusters muß die Rechenkapazität eines einzelnen Prozessors sorgfältig auf die Applikationen verteilt werden.

[0024] Dazu muß das Betriebssystem einer jeden Plattform die Prozesse, die auf ein und derselben Plattform um Rechenzeit konkurrieren, besonders unter hoher Last so unterstützen, daß sie die Realzeitanforderungen an den Vermittlungsknoten einhalten können. Um das zu ermöglichen, werden die Prozesse zu Aufgabengruppen zusammengefaßt.

[0025] FIG 2 zeigt ein Beispiel zum Scheduling der Prozesse. Für jede der Aufgabengruppen ist durch Laststeuerungsparameter ein garantierter Mindestanteil der Prozessorzeit als Laufzeitbudget (d.h. Nettorechenzeit) definiert. Daher werden diese Aufgabengruppen auch als Virtuelle CPU (VCPU) bezeichnet. Pro VCPU und Priorität werden die Prozesse einer FIFO-Warteschlange zugewiesen und warten dort auf Zuteilung der CPU.

[0026] Das Scheduling arbeitet im Time Slicing. Zu Beginn eines jeden Zeitschlitzes betrachtet es pro VCPU ein Zeitguthabenkonto, indem es die Summe aller gemessenen Laufzeiten der Prozesse einer VCPU mit dem garantierten Zeitbudget vergleicht. Das Resultat dieses Vergleichs ist ein VCPU-individuelles Zeitguthaben.

[0027] In der VCPU mit dem größten Zeitguthaben bekommt derjenige wartende Prozeß als erster Rechenzeit, der an der ersten Stelle der Warteschlange höchster Priorität wartet (siehe FIG. 2).

[0028] Veragt sich der Prozeß noch während diese Zeitschlitzes, so folgt aus der gleichen VCPU (und gegebenenfalls der gleichen Warteschlange) der nächste Prozeß. Wartet aus dieser VCPU kein Prozeß mehr, so kommt der Scheduling-Algorithmus wie zu Beginn des Zeitschlitzes zum Tragen. Zu Beginn des nächsten Zeitschlitzes findet erneut der Schedulingalgorithmus Anwendung.

Lastkontrolle (Laststeuerung)

[0029] Die Lastkontroll-SW vergleicht zyklisch die aktuelle Lastsituation der eigenen Plattform mit plattformindividuellen Schwellenwerte und kennzeichnet gegebenenfalls den Prozessor als im Zustand Überlast befindlich. Gegeben ist der Zustand Überlast für einen Prozessor, wenn die angebotene totale Last höher liegt, als die Lastschwellen, die für diesen Prozessor gegeben sind. Diese Lastinformation wird systemweit verteilt, so daß Service Adressing in der Lage ist, überlastete Plattformen zu "schonen".

[0030] Sobald ein Prozessor im Zustand Überlast ist, stellt die Lastkontrolle außerdem anhand der Laufzeitbudgets der VCPU fest, welche dieser VCPU ihr Budget überschritten hat/haben. Diese VCPU/n wird/werden dann auch als überlastet bezeichnet. Geeignete Prozesse innerhalb dieser überlasteten VCPU/n werden informiert und damit aufgefordert, ihre Lastangebot an den Prozessor zu reduzieren.

[0031] Nach der oben dargelegten MP-Betriebssystemstruktur des ATM Vermittlungsknotens arbeiten also sowohl die Laststeuerung als auch das Prozeß-Scheduling im Betriebssystem mit einer Reihe von Laststeuerungsparametern, um die Voraussetzungen für ein optimales Realzeitverhalten zu schaffen. Da Betriebssystem und Laststeuerung für alle MP-Plattformen generisch, d.h. einheitlich sind, müssen diese Daten für sämtliche möglichen Funktionsausprägungen einer Plattform auf jedem MP verfügbar sein. Außerdem müssen sie interaktiv auf der Plattform "aktivierbar" sein, z.B. während des Konfigurierens, um der Funktionsausprägung Rechnung zu tragen. Des weiteren müssen die Daten dem jeweiligen Betriebszustand (z.B. Recovery) der Plattform angepaßt sein, um auf die verschiedenen Anforderungen dieser Betriebszustände optimal einzugehen.

Load Model Management

[0032] Die dargestellten Anforderungen und Mechanismen, wie Skalierbarkeit des Netzknotens, Verteilung des Lastangebots auf die Plattformen, Aufteilung der Rechnerkapazität einer Plattform auf die verschiedenen Anwender, Prozeßkonzept mit statischen Prozeßattri-

buten sowie die Lastkontrolle sind noch nicht alle Mechanismen, die im Zusammenhang mit Laststeuerungsparametern stehen. Darüberhinaus gibt es noch Taktinformationen, Laufzeitbeschränkungen für das interne Protokoll, die Berücksichtigung der Grundlast, weitere Unter-Budgetaufteilungen und Schwellenwerte, auf die hier nicht eingegangen werden soll.

[0033] Betrachtet man Menge und Komplexität der Laststeuerparameter, sowie Anforderungsprofil und Funktionsgeflecht, in dem sie stehen, so wird deutlich, daß es notwendig ist, sie übersichtlich zu implementieren und von einer eigenen Software kontrollieren zu lassen.

[0034] Alle Parameter, die die Laststeuerung und die Lastverteilung des Systems beeinflussen, werden deshalb in einem auf jeder Plattform replizierten Katalog abgespeichert. Das schafft eine Übersichtlichkeit, die insbesondere wichtig ist, wenn in der Designphase die Änderung an einem der Parameter mit den Werten anderer Parameter interferiert. Zur Laufzeit hat nur noch eine einzige Managementfunktion (LM-Manager) Zugang zu diesem Katalog, die auch die Schnittstellen zu den Parameteranwendern unterhält. Diese Managementfunktion ist auch empfänglich für Trigger, die über definierte Schnittstellen zur Laufzeit Änderungen der aktuellen gültigen Parameter anfordern können (z.B. über interaktive Kommandos durch den Betreiber). Sie trägt die Verantwortung für das Informieren aller von den Änderungen betroffenen Anwender. Die Struktur des Parameterkatalogs ist so gewählt, daß online-Änderungen der gültigen Parameter sinnvoll eingeschränkt werden. Nur durch diese Form des Managements und der Strukturierung ist ein Engineering der komplexen Einflußgrößen möglich und fehlerfrei implementierbar.

[0035] Durch das Load Model Management wird die komplexe und umfangreiche Vielfalt von Laststeuerungsparametern in einer Weise strukturiert, die zum einen die Übersichtlichkeit zum Zwecke der Engineerbarkeit gewährleistet, einschließlich der fehlerfreien Implementierbarkeit. Zum anderen wird aber eine Flexibilität ermöglicht, die Voraussetzungen für eine situationsgerechte Anpassung (bei der Konfiguration und zur Laufzeit) der aktuell gültigen Parameterwerte zum Zwecke der optimalen Performance hinsichtlich Lastverteilung und Laststeuerung.

[0036] Interaktiv kann aus dem Parameterkatalog immer nur eine einzige Tabelle ausgewählt und "aktiviert" werden, wodurch kein Wert aus einer anderen Tabelle des Katalogs Anwendung finden kann (solange diese Tabelle "aktiv" ist). Wechselnde Betriebszustände der Plattform wiederum können nur zwischen vorgefertigten Datensätzen auswählen und zwischen diesen wechseln.

[0037] FIG 3 zeigt die Beziehungen des Lastmodell-Managers zu verschiedenen Komponenten innerhalb des Systems. Es gibt drei unterschiedliche Typen von Anwendern des LM-Managers. Einen ersten Anwender-

typ (Anwender l), der nur mit dem LM-Manager in Verbindung steht. Als beispielh. Anwender ist hier die Recovery-Komponente zu nennen. Einen zweiten Anwendertyp (Anwender i), der sowohl mit dem LM-Manager als auch mit der Laststeuerung in Verbindung steht. Als beispielh. Anwender ist hier die Scheduling-Komponente zu nennen. Schließlich dritten Anwendertyp (Anwender n), der nur Werte von der Laststeuerung erhält. Als beispielh. Anwender ist hier die Vermittlungstechnik-Komponente zu nennen.

Betreiberschnittstelle

[0038] Z.B. bei der Inbetriebnahme des Vermittlungsknotens muß jede einzelne Plattform des Clusters konfiguriert werden. Das bedeutet,

☑ daß der Ladetyp zugewiesen und geladen wird und

☑ daß aus dem auf allen Plattformen replizierten Lastmodellkatalog eine Lastmodelltabelle ausgewählt wird.

Die Art der Organisation der zentralen Laststeuerparameter im Lastmodellkatalog wird unten im Abschnitt "Lastmodellkatalog" vertieft.

[0039] Eine Lastmodelltabelle aus dem Lastmodellkatalog ist einem Ladetyp eindeutig zugeordnet, wobei für einen Ladetyp mehrere Tabellen vorhanden sein können (z.B. für unterschiedliche Belastungserwartungen oder Betriebszustände).

[0040] Nach der Auswahl der Tabelle gelten dann bis zum unterbrechungsfreien Austausch dieser Tabelle ausschließlich die in ihr aufgeführten Werte, die einen in sich konsistenten Satz an Parametern darstellen. Das bedeutet eine sinnvolle Einschränkung, die dem Betreiber die Bedienung des Knotens erleichtert und ihm die Sicherheit gibt, immer aufeinander abgestimmte Parameter in Betrieb zu nehmen. Die Auswahlmöglichkeit unter verschiedenen Lastmodelltabellen gibt ihm die Möglichkeit eines Tunings der Plattformen.

[0041] Das heißt, der Betreiber kann je nach Belastungserwartung an die verschiedenen VCPUn geeignete VCPU-Budgets einrichten. Da er unterbrechungsfrei während des Betriebs die aktuell gültige Lastmodelltabelle austauschen kann, können die Budgets problemlos geänderten Bedingungen (z.B. durch HW- oder SW-Upgrade im Knoten) angepaßt werden.

[0042] Per SW-Patch, der vom Hersteller auf Konsistenz geprüft und geliefert werden kann, können Werte innerhalb einer Tabelle geändert werden. Ein Betreiberkommando kann dann die gepatchte Tabelle betriebsunterbrechungsfrei selektieren (vgl. Abb. 3) und die Steuerungssoftware (Load Model Management) sorgt für die Information der betroffenen Applikationen.)

[0043] Zur Koordination der Auswahl siehe weiter unten den Abschnitt "Steuerungssoftware" und FIG. 4.

Lastmodellkatalog

[0044] Der Lastmodellkatalog stellt die Organisationsstruktur derjenigen Lastmodellparameter dar, die zentral implementiert sind.¹

[0045] Er besteht aus einzelnen Lastmodelltabellen, die interaktiv jeder Plattform zugewiesen werden können. (vgl. Abschnitt "Betreiberschnittstelle".)

[0046] Jede Lastmodelltabelle aus dem Lastmodellkatalog enthält zum einen mehrere Lastmodelle, die die zu garantierenden Budgets für die VCPUn enthalten. Verschiedene Lastmodelle werden benötigt, um den unterschiedlichen Anforderungen der Plattform-Betriebszustände, wie z.B. Startup oder Normalbetrieb, Rechnung tragen zu können.

[0047] Die Lastmodelltabelle enthält zum anderen noch die Taktinformationen, Überlastgrenzen und Laufzeitbeschränkungen für die Abwicklung des internen Protokolls, die Größe der Lastreserve, die Basislast sowie eine Tabelle, die die Zuordnung der Prozesse auf die VCPUn unterstützt.

Steuerungssoftware (Load Model Manager)

[0048] Die Steuerungs-SW (Load Model Management bzw. Management-Komponente) ist ein betriebssystemnaher Prozess, der für das Auslesen der richtigen Parameter-Daten bei Inbetriebnahme bzw. Plattformzustandsänderungen sorgt.

[0049] Der Load Model Manager (Kurzform: LM-Manager) ermöglicht den unterbrechungsfreien Austausch der aktuell gültigen Lastmodellparameter. Dazu stellt er Schnittstellen für die triggernden Ereignisse zur Verfügung. Des weiteren sorgt er für die Information der von diesem Ereignis betroffenen SW. Zu dieser SW gehören das Betriebssystem sowie Prozesse wie z.B. die Lastkontrolle. Wird durch einen Austausch der Lastmodelltabelle einer VCPU das Budget völlig entzogen, so müssen natürlich auch die dazugehörigen Service zurückgezogen werden.

[0050] Je nach internem Betriebszustand der betreffenden Plattform findet das eine oder andere Lastmodell Anwendung. Findet z.B. (siehe auch FIG 3 oder 4) während des Betriebs ein Recovery statt, so kann Recovery sich an Load Management wenden, damit vom normalen Betriebslastmodell auf ein spezielles Recovery-Lastmodell umgeschaltet wird, so daß den geänderten Laufzeitanprüchen von Recovery Rechnung getragen wird. Load Model Management informiert die davon betroffene SW, wie z.B. Scheduling und die Lastkontrolle, selbständig von diesem Ereignis.

¹ Im Gegensatz dazu sind Prozeßattribute ja dezentral bei jedem einzelnen Prozeß erklärt, vgl. Oben den Abschnitt "Prozeßbegriff".

Abkürzungen

[0051]

- ATM	Asynchronous Transfer Mode	5
- CCITT	Comitee Consultatif de Telephonie et Telegraphie (heute ITU-T)	
- CCS7	Common Channel Signalling System No.7 (Trunksignallisierung feste Bandbreite)	10
- CHILL	CCITT High Level Language	
- FIFO	First In First Out	
- ITU	International Telecommunication Union	
- Lmo oder LM	Lastmodell (Load Model)	15
- NNI	Network Network Interface ("Trunk"signallisierung variable Bandbreite)	
- PNNI	Private Network Network Interface	
- SW	Software	20
- UNI	User Network Interface (Breitband Teilnehmer Signallisierung)	
- VCPU	Virtuelle CPU	

Patentansprüche

1. Plattform eines Rechnersystems, **gekennzeichnet durch** eine Manager-Komponente, den sogenannten Last-Modell-Manager, die Parameter für die Lastverteilung und Laststeuerung des Systems einheitlich verwaltet. 30
2. Plattform nach Anspruch 1, **dadurch gekennzeichnet, daß** der genannte Lastmodell-Manager die genannten Parameter in einem Katalog verwaltet, der mindestens eine Tabelle umfaßt, wobei eine Tabelle jeweils ein Lastmodell umfaßt, d.h. einen kompletten konsistenten Satz der Parameter, die die Lastverteilung und -steuerung des Systems beeinflussen. 35 40
3. Plattform nach Anspruch 1 oder 2, **dadurch gekennzeichnet, daß** der Lastmodell-Manager eine Tabelle des genannten Katalogs als aktuell gültige Tabelle bestimmt. 45
4. Plattform nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet, daß** der Lastmodell-Manager mindestens eine Schnittstelle umfaßt, über die eine Bestimmung der aktuell gültigen Tabelle während des Betriebs getriggert werden kann. 50 55
5. Plattform nach einem der Ansprüche 3 oder 4, **dadurch gekennzeichnet, daß**

die durch die genannte Bestimmung vollzogene Auswahl vom Ladetyp einer Plattform abhängt.

6. Plattform nach einem der Ansprüche 3 bis 5, **dadurch gekennzeichnet, daß** die durch die genannte Bestimmung vollzogene Auswahl vom Betriebszustand einer Plattform abhängt.
7. Plattform nach einem der Ansprüche 3 bis 6, **dadurch gekennzeichnet, daß** die durch die genannte Bestimmung vollzogene Auswahl von einer Betreibereingabe abhängt.
8. Plattform nach einem der Ansprüche 2 bis 7, **dadurch gekennzeichnet, daß** der Lastmodell-Manager Schnittstellen zu SW-Komponenten der Plattform aufweist, über die er diesen SW-Komponenten aktuell gültige Laststeuerungs-Parameter aus der aktuell gültigen Tabelle mitteilt.
9. Plattform nach einem der Ansprüche 2 bis 8, **dadurch gekennzeichnet, daß** der Lastmodell-Manager eine Schnittstelle umfaßt, über die der Systemhersteller den Inhalt einer Tabelle während des Betriebs ändern kann.

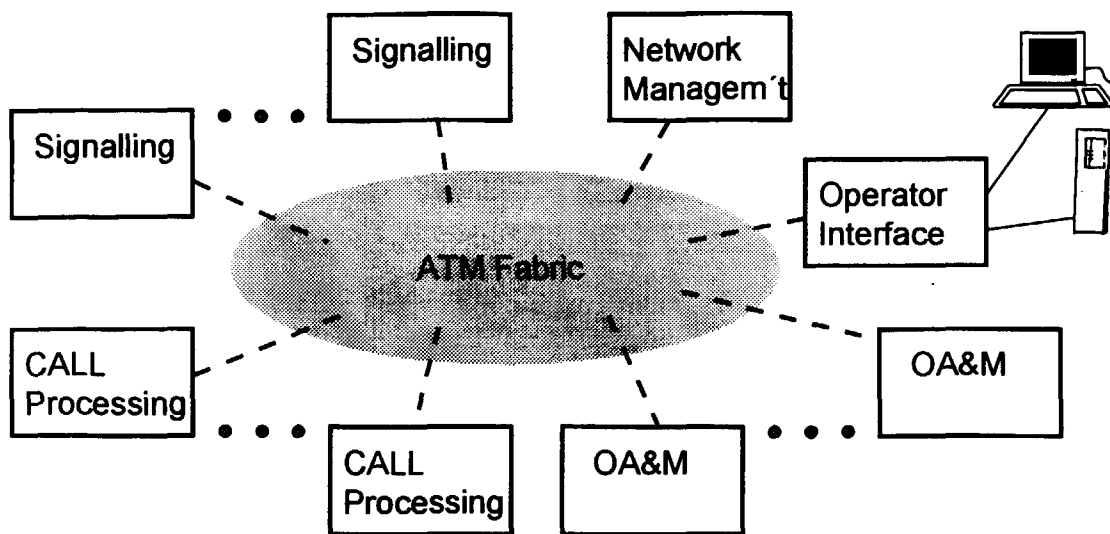
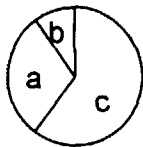


Abbildung 1 Cluster der Hauptprozessoren, große Konfiguration

FIG.1

VCPU Budgets [%] für die VCPUn a, b, c
(Lastmodell)



davon verbrauchte Laufzeit a', b', c'
und verbleibendes Zeitguthaben für
die VCPUn a, b, c

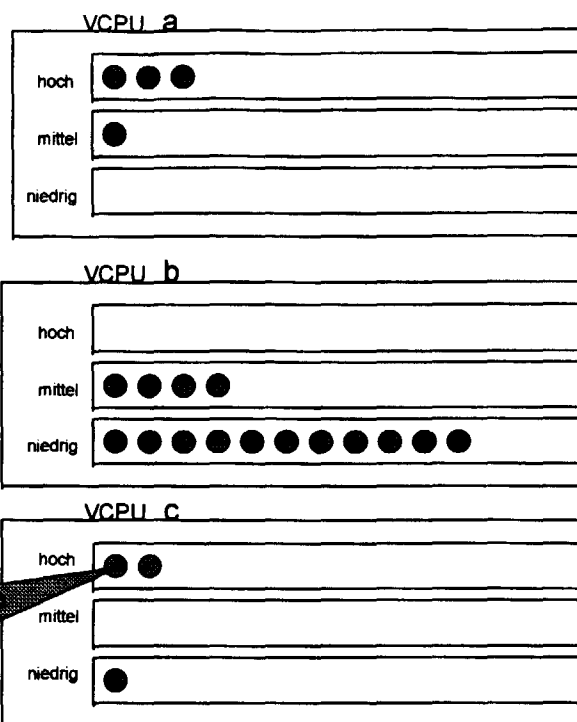
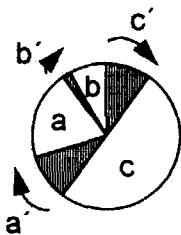


Abbildung 2:
Beispiel zum Scheduling der Prozesse: VCPU-Zeitguthaben und Prozeßwarteschlangen nach VCPUn und Prioritäten

FIG.2

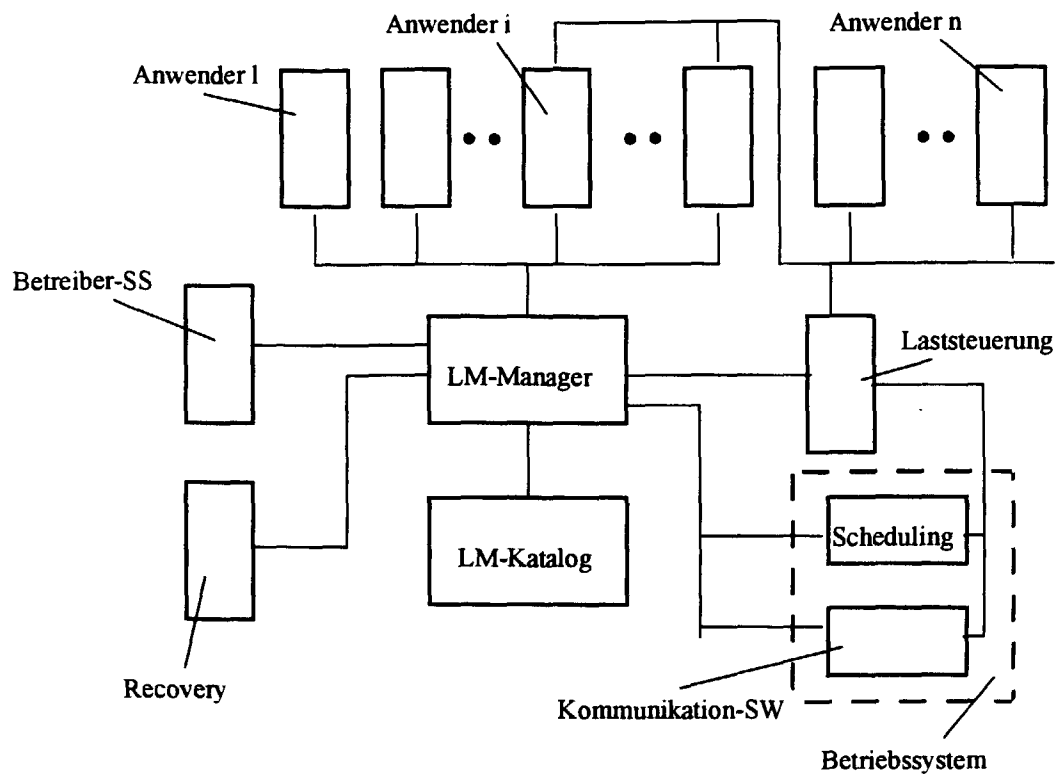


FIG 3

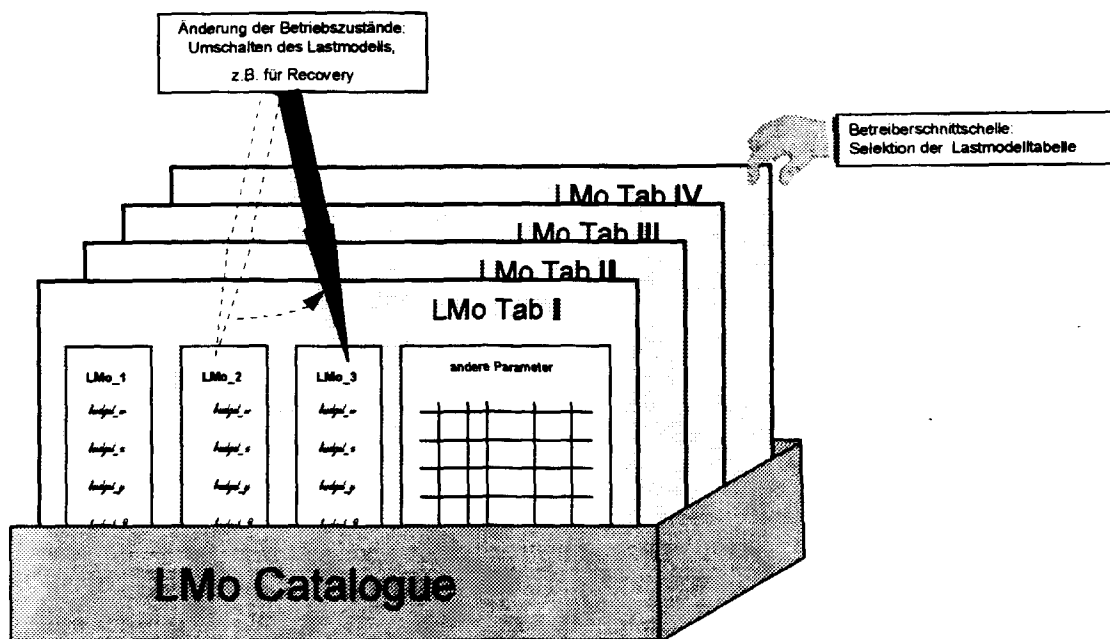


Abbildung 3 Lastmodellkatalog

FIG 4



Europäisches
Patentamt

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung
EP 98 10 3827

EINSCHLÄGIGE DOKUMENTE			
Kategorie	Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile	Betrifft Anspruch	KLASSIFIKATION DER ANMELDUNG (Int.Cl.6)
A	EP 0 346 039 A (DEMAX SOFTWARE INC) 13. Dezember 1989 * das ganze Dokument * ---	1-9	G06F9/46
A	US 5 537 596 A (DEROSSO CHRISTOPHER S ET AL) 16. Juli 1996 * das ganze Dokument * -----	1	
			RECHERCHIERTE SACHGEBIETE (Int.Cl.6)
			G06F
Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt			
Recherchenort DEN HAAG		Abschlußdatum der Recherche 11. August 1998	
		Prüfer Michel, T	
KATEGORIE DER GENANNTEN DOKUMENTE X : von besonderer Bedeutung allein betrachtet Y : von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie A : technologischer Hintergrund O : mündliche Offenbarung P : Zwischenliteratur T : der Erfindung zugrunde liegende Theorien oder Grundsätze E : älteres Patentedokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist D : in der Anmeldung angeführtes Dokument L : aus anderen Gründen angeführtes Dokument & : Mitglied der gleichen Patentfamilie, übereinstimmendes Dokument			

EPO FORM 1503 03/82 (P04C03)